Fleißige Biber

In den frühen sechziger Jahren ging Tibor Rado von der Ohio State University der Frage nach, wie viele Einsen eine Turingmaschine wohl auf ein zu Beginn leeres Band schreiben könne, ehe sie anhält.

Genauer formuliert lautet das Problem:

Eine Turingmaschine mit n inneren Zuständen beginnt ihre Arbeit auf einem (leeren) Band, das nur Nullen (bzw. BLANKS) enthält. Wie viele Einsen kann sie dann auf das Band schreiben, bevor sie schließlich (irgendwann) anhält?

Kurz: Welches ist die fleißigste unter allen erdenklichen Turingmaschinen mit n Zuständen ?

Für n = 2, n = 3, n = 4 und n = 5 ist die Antwort auf diese Frage inzwischen bekannt, nicht jedoch für n = 6 oder noch höhere n.

Die rastlose Geschäftigkeit der Biber (lat.: castor) war zweifellos der Grund, dass T. Rado diesen speziellen **Turingmaschinen** den Namen

Busy- Beaver (lat.: castor sedulus)

gab.

Im Jahr 1983 fand in Dortmund ein Biber-Wettbewerb statt mit dem Ziel, den fleißigen Biber mit n=6* Zuständen zu finden. Auf der Suche nach geeigneten Kandidaten für diesen Wettbewerb hatten die Teilnehmer monatelang Programme zum Generieren von Turingmaschinen entworfen, bzw. Turingmaschinen und ihr Verhalten theoretisch analysiert.
Der Deutsche Teilnehmer Uwe Schult hatte dafür eine spezielle Hardware zum Testen der Maschinen entwickelt. Die technische Leistungsfähigkeit der Hardware zu dieser Zeit machte es notwendig, bei dem Test der Turingmaschinen folgende Einschränkungen zu machen:

- Die Länge des Bands war auf 4096 Felder beschränkt
- Die maximale Schrittzahl war auf 500000 beschränkt

* Bei vielen Autoren wird der Endzustand HALT nicht mitgezählt, sodass hier nur 5 Zustände gezählt werden.

Das Ergebnis seiner Arbeit war ein Biberkandidat, der 501 Einsen auf das Band schrieb, bevor er anhielt. Nebenbei wurden noch eine Reihe neuer, absonderlicher Biberabkömmlinge entdeckt, die keine Eins auf dem Band hinterlassen.

Definition:

Ein fleißiger Biber mit n Zuständen ist eine Turingmaschine mit n Zuständen, die zwei Bedingungen erfüllt:

1) Sie bleibt, wenn man sie auf ein Band mit lauter Nullen(bzw. BLANKS) ansetzt, irgendwann stehen.
2) Sie schreibt zwischen Start und Stopp mindestens so viele Einsen auf das Band, wie jede andere Turingmaschine mit n Zuständen, die gleichfalls anhält.

Die maximale Zahl von Einsen, die eine irgendwann anhaltende Turingmaschine mit n Zuständen auf ein Band mit lauter Nullen druckt, bezeichnet man als B(n).

B ist dabei die sogenannte Biberfunktion bzw. Radofunktion
Bis heute sind lediglich vier Funktionswerte bekannt:

B(2)=1 B(3)=4 B(4)=6 und B(5)=13

Den fleißigen Biber mit vier Zuständen haben im Jahre 1962 Rado und Shen Lin von den AT&T Bell Laboratories entdeckt. Er schreibt in 13 Schritten 6 Einsen auf das Band.

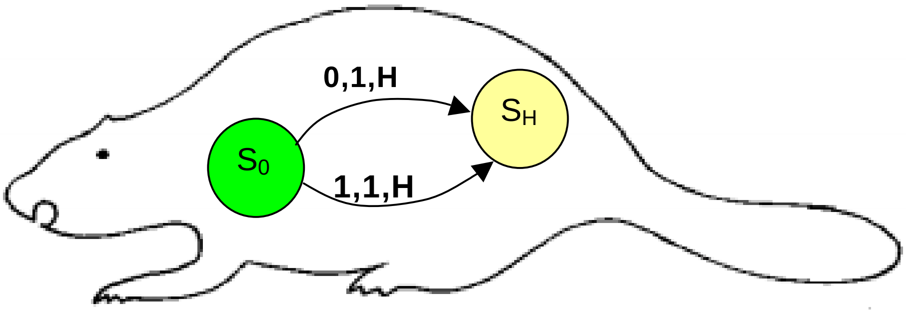
Erst 1973 fand Bruno Weimann von der Universität Bonn den fleißigen Biber mit fünf Zuständen: Er produziert in 107 Schritten dreizehn Einsen.

Der fleißige Biber mit 2 Zuständen:

$T_2 = (\Sigma;S;B;F;s_0;\varphi_2)$

$\Sigma = \{0;1\}$ $S = \{s_0,s_H\}$ $B = \{R,L,H\}$ $F=\{S_H\}$ $s_0 \in S$ $\varphi:(e_i;S_j) \to (e_k;S_r;b_s)$

φ_2 :



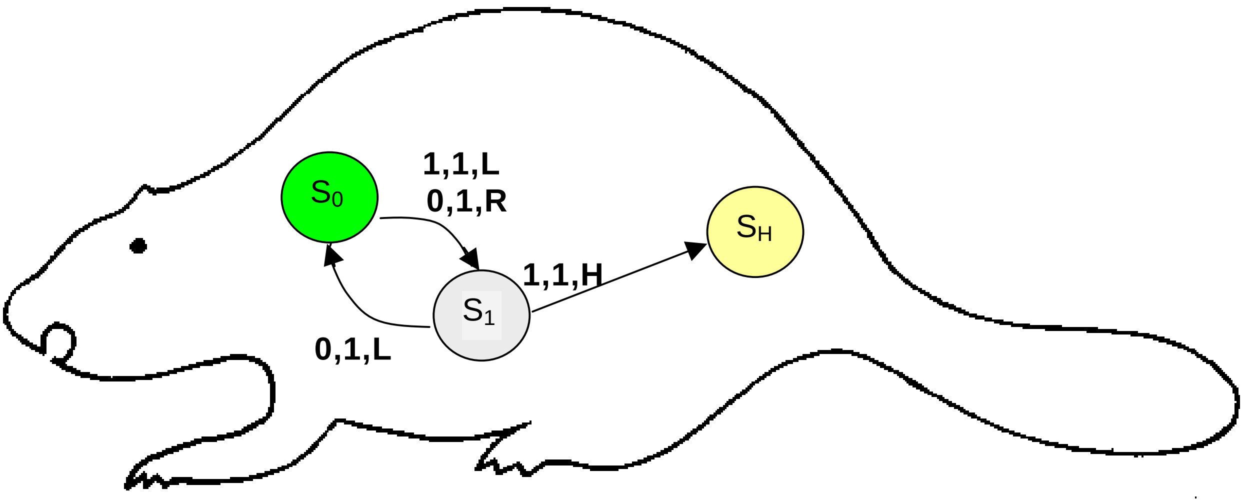
Er Produziert in einem Schritt eine 1 und bleibt dann stehen.

Der fleißige Biber mit 3 Zuständen:

$T_3 = (\Sigma;S;B;F;s_0;\varphi_3)$

$\Sigma = \{0;1\}$ $S = \{s_0,s_1,s_H\}$ $B = \{R,L,H\}$ $F=\{S_H\}$ $s_0 \in S$ $\varphi_3:(e_i;S_j) \to (e_k;S_r;b_s)$

φ_3 :



Ablaufprotokoll:

Bandbeschriftung										Zustand
0	0	0	0	0	0	0	0	0	0	S_0
0	0	0	0	0	1	0	0	0	0	S_1
0	0	0	0	0	1	1	0	0	0	S_0
0	0	0	0	0	1	1	0	0	0	S_1
0	0	0	0	1	1	1	0	0	0	S_0
0	0	0	1	1	1	1	0	0	0	S_1
0	0	0	1	1	1	1	0	0	0	S_H

Der Biber T_3 schreibt in 6 Schritten 4 Einsen kompakt nebeneinander auf das Band. Der Schreib-Lesekopf steht am Ende wieder auf der Startposition.

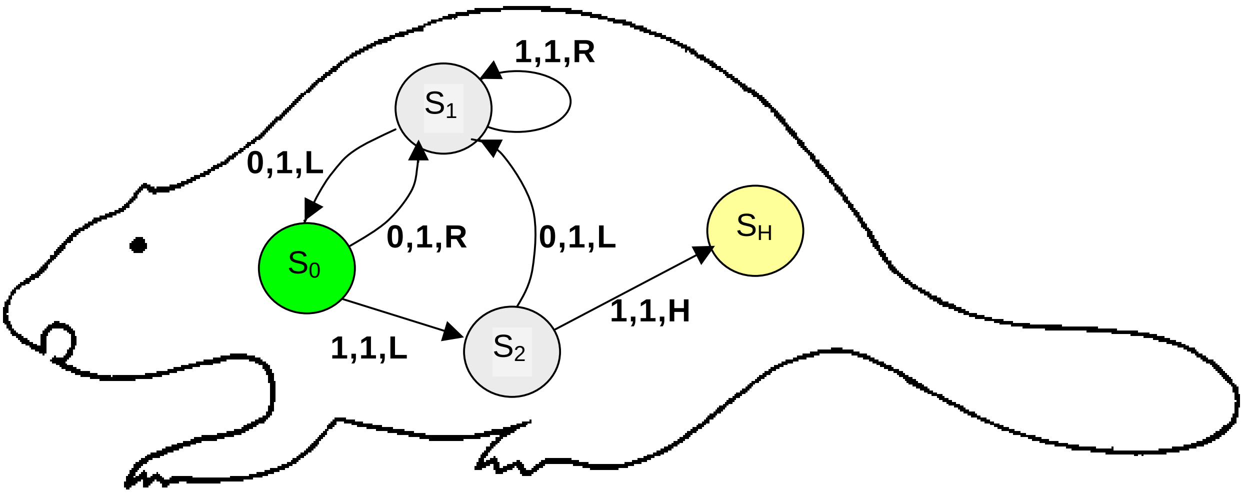
Der fleißige Biber mit 4 Zuständen:

$T_4 = (\Sigma;S;B;F;s_0;\varphi_4)$

$\Sigma = \{0;1\}$ $S = \{s_0,s_1,s_2,s_H\}$ $B = \{R,L,H\}$ $F=\{S_H\}$ $s_0 \in S$

$\varphi_4 : (e_i;S_j) \rightarrow (e_k;S_r;b_s)$

φ_4 :



Ablaufprotokoll:

Bandbeschriftung										Zustand
0	0	0	0	0	0	0	0	0	0	S_0
0	0	0	0	0	1	0	0	0	0	S_1
0	0	0	0	0	1	1	0	0	0	S_0
0	0	0	0	0	1	1	0	0	0	S_2
0	0	0	0	1	1	1	0	0	0	S_1
0	0	0	1	1	1	1	0	0	0	S_0
0	0	1	1	1	1	1	0	0	0	S_1
0	0	1	1	1	1	1	0	0	0	S_1
0	0	1	1	1	1	1	0	0	0	S_1
0	0	1	1	1	1	1	0	0	0	S_1
0	0	1	1	1	1	1	0	0	0	S_1
0	0	1	1	1	1	1	1	0	0	S_0
0	0	1	1	1	1	1	1	0	0	S_2
0	0	1	1	1	1	1	1	0	0	S_H

Dieser Biber schreibt in 13 Schritten 6 Einsen auf das leere Band.

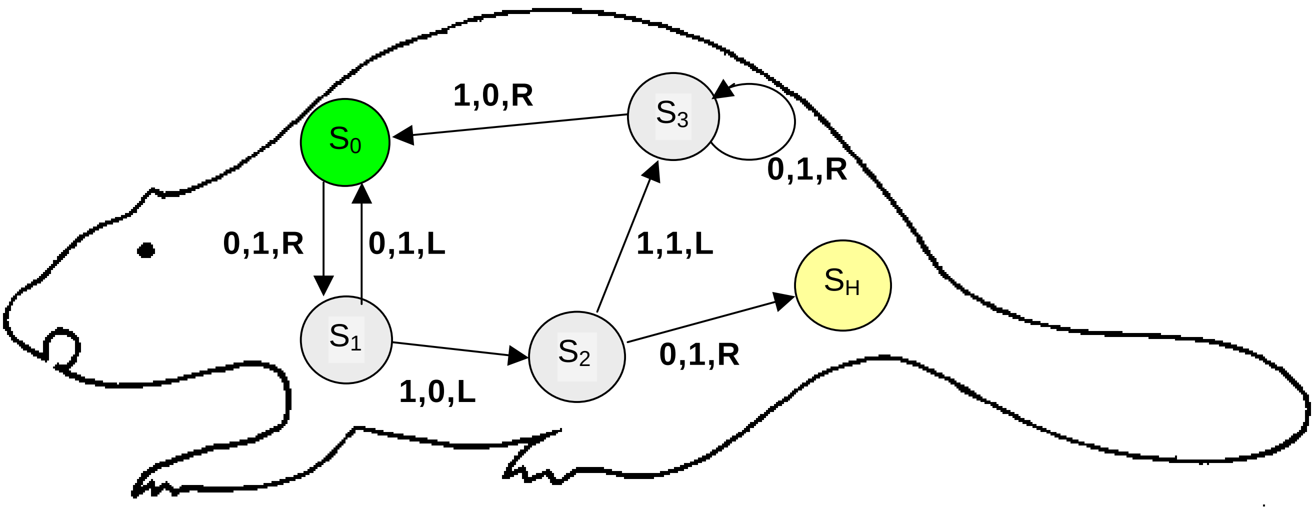
Der fleißige Biber mit 5 Zuständen:

$T_5 = (\Sigma;S;B;F;s_0;\varphi_5)$

$\Sigma = \{0;1\}$ $S = \{s_0, s_1, s_2, s_3, s_H\}$ $B = \{R,L,H\}$ $F = \{S_H\}$ $s_0 \in S$

$\varphi_5 : (e_i;S_j) \to (e_k;S_r;b_s)$

φ_5 :



Ablaufprotokoll: Dieser Biber schreibt in 107 Schritten 13 Einsen auf das leere Band.

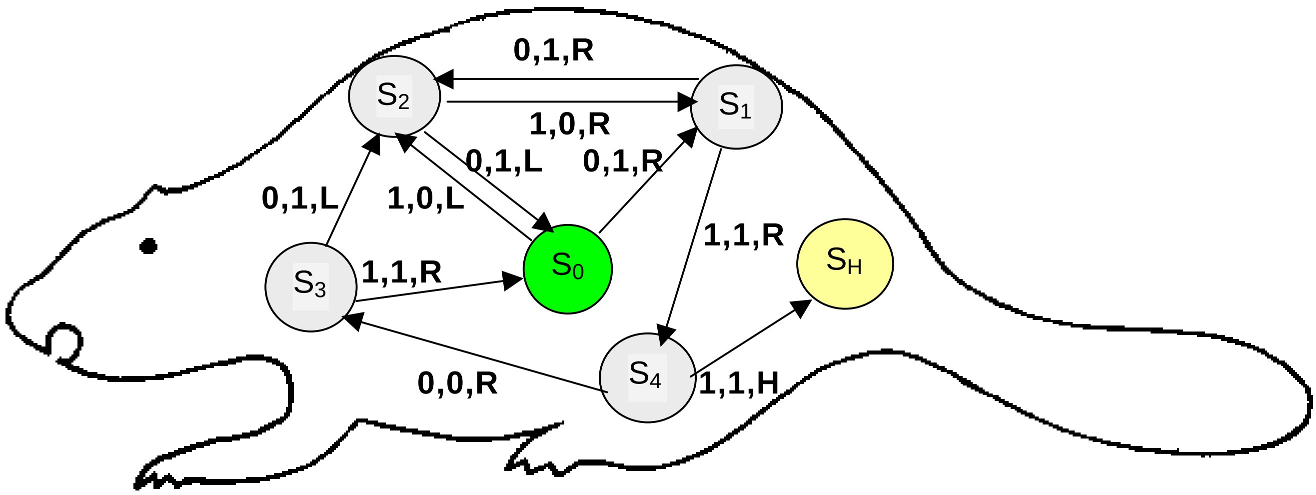
Biberkandidaten mit 6 Zuständen

$T_{61} = (\Sigma;S;B;F;s_0;\varphi_{61})$ Uwe Schult 1983

$\Sigma = \{0;1\}$ $S = \{s_0,s_1,s_2,s_3,s_4,s_H\}$ $B = \{R,L,H\}$ $F=\{S_H\}$ $s_0 \in S$

$\varphi_{61} : (e_i;S_j) \rightarrow (e_k;S_r;b_s)$

φ_{61}:



Dieser Biberkandidat schreibt 501 Einsen auf das leere Band und benötigt dabei 134467 Schritte.

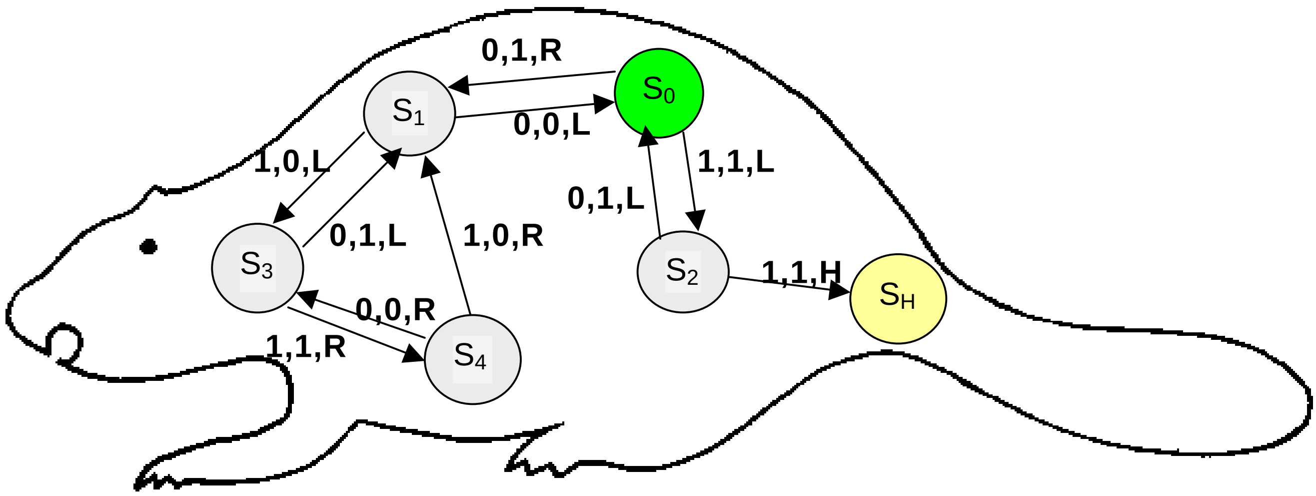
Biberkandidaten mit 6 Zuständen

$T_{62} = (\Sigma;S;B;F;s_0;\varphi_{62})$ George Uhing N.Y. 1984

$\Sigma = \{0;1\}$ $S = \{s_0, s_1, s_2, s_3, s_4, s_H\}$ $B = \{R,L,H\}$ $F=\{S_H\}$ $s_0 \in S$

$\varphi_{62} : (e_i;S_j) \rightarrow (e_k;S_r;b_s)$

φ_{62} :



Dieser Biberkandidat schreibt 1913 Einsen auf das leere Band und benötigt dabei 2133492 Schritte.

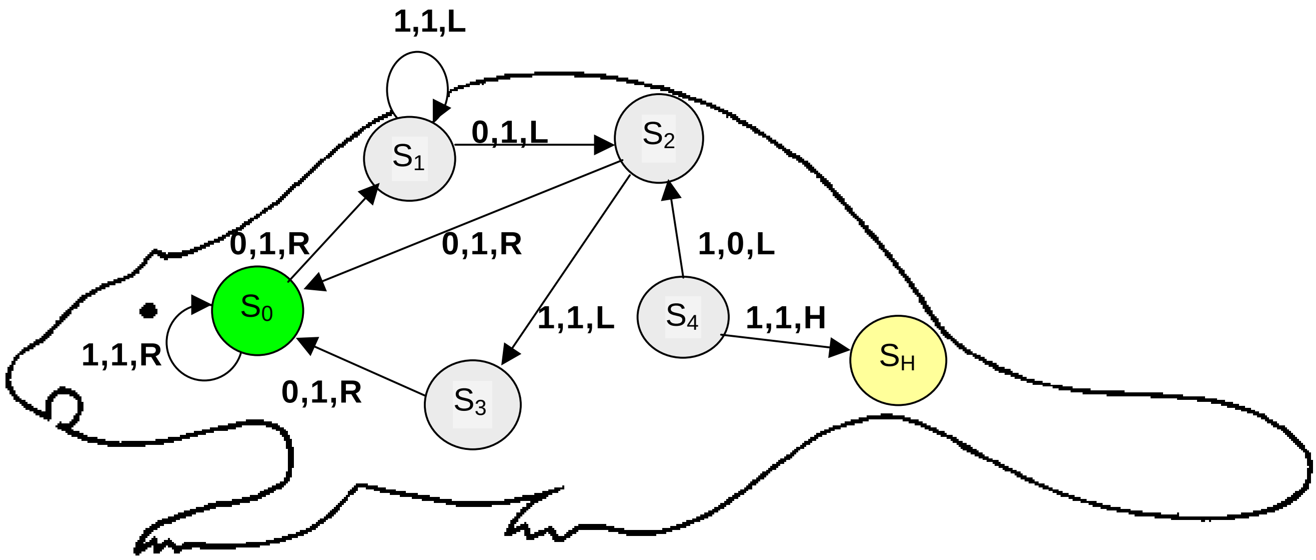
Biberkandidaten mit 6 Zuständen

$T_{63} = (\Sigma;S;B;F;s_0;\varphi_{63})$ H.Marxen/J.Buntrock 1989

$\Sigma = \{0;1\}$ $S = \{s_0,s_1,s_2,s_3,s_4,s_H\}$ $B = \{R,L,H\}$ $F=\{S_H\}$ $s_0 \in S$

$\varphi_{63} : (e_i;S_j) \to (e_k;S_r;b_s)$

φ_{63} :



Dieser Biberkandidat schreibt 4098 Einsen auf das leere Band und benötigt dabei 47176870 Schritte.

Er ist bis heute (2001) der Rekordhalter.

Ein einfacher „Algorithmus“ zur Berechnung von B(n) könnte so aussehen:

1.	Eingabe n
2.	Notiere alle Turingmaschinen mit n Zuständen
3.	Sortiere alle nicht anhaltenden Maschinen aus
4	Starte jede haltende Maschine auf einem leeren Band
4.	Notiere die Anzahl der Einsen für jede Maschine
5.	Das Maximum der erhaltenen Ergebnisse ist B(n)

Problem:

- o Wie viele Turingmaschinen mit n Zuständen gibt es ?
- o Gibt es einen Algorithmus, mit dem man entscheiden kann, ob eine Turingmaschine anhält ? HALTEPROBLEM

Die Anzahl der TM mit n Zuständen lässt sich leicht ermitteln:
Die Turingmaschinen unterscheiden sich nur durch ihre Überführungsfunktionen, die in Tabellenform folgendermaßen aussehen:

S_0	0			
S_0	1			
S_1	0			
S_1	1			
S_2	0			
S_1	1			
..				
..				
S_{n-1}	0			
S_{n-1}	1			

S_k	e_i	S_r	e_m	k

- aktueller Zustand
- aktuell gelesenes Zeichen
- Folge-zustand
- rückgeschrie-benes Zeichen
- Kopf - bewegung

Es gibt insgesamt 2*(n-1) Tripel, die noch ausgefüllt werden können:

S_r	e_m	k

Da die Kopfbewegung H nur beim Übergang in den Endzustand S_H vorkommt betrachtet man zunächst die Anzahl der Möglichkeiten ein Tripel ohne die Kopfbewegung H auszufüllen:

N = (n-1)* 2 * 2

Mit H gibt es zusätzlich

S_H	e_m	H

M = 1 * 2 * 1

Also gibt es insgesamt $2\cdot(n-1)\cdot 2+1\cdot 2\cdot 1=4n-2$ Möglichkeiten jedes Tripel in der obigen Tabelle auszufüllen.
Damit gibt es dann $(4n-2)\cdot(4n-2)\cdot\cdots\cdot(4n-2)=(4n-2)^{2n-2}$ mögliche Turingmaschinen mit n Zuständen.

n	Anzahl der TM mit n Zuständen
2	$6^2=36$
3	$10^4=10000$
4	$14^6=7529536$
5	$18^8=11.019.960.576$
6	$22^{10}=26.559.922.791.424$
7	$26^{12}=95.428.956.661.682.176$

Zum Vergleich: 1y=31536000s $\frac{22^{10}}{31536000}\approx 8{,}4\cdot 10^5$

D.h. Wenn es gelingt, jede Sekunde eine TM mit 6 Zuständen zu generieren und auszutesten, so würde das ca. 840.000 Jahre benötigen!